IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Mark Nathan Wain, et al.**	Confirmation No.: **1285**
Application No.: **10/678,870**	Group Art Unit: **2192**
Filing Date: **October 3, 2003**	Examiner: **Isaac Tuku Tecklu**
For: **Extensible Multi-Language Compilation**	

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

SUPPLEMENTAL REPLY PURSUANT TO 37 CFR § 1.111

In further response to the Official Action dated **January 8, 2009**, reconsideration is respectfully requested in view of the amendments and/or remarks as indicated below:

☐ **Amendments to the Specification** begin on page of this paper.

☒ **Amendments to the Claims** are reflected in the listing of the claims which begins on page of this 2 paper.

☐ **Amendments to the Drawings** begin on page of this paper and include an attached replacement sheet.

☒ **Remarks** begin on page 12 of this paper.

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) A method for multi-language compilation comprising:

discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol and provides compilation of code in a second language, wherein the discovery of the first compiler component and the second compiler component is effected using an inter-process communication; and

providing a first framework which uses said first compiler component and said second compiler component to provide multi-language compilation, wherein the framework integrates the first compiler component and the second compiler component to create a multi-language compiler,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first compiler component and said discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second compiler component.

2. (Original) The method of claim 1, further comprising:

discovering a third compiler component which implements said prespecified protocol; and

providing a second framework which integrates said first, second, and third compiler components to provide multi-language compilation.

3. (Canceled)

4. (Original) The method of claim 1, where said prespecified protocol comprises a compilation time interface, and where said provision of multi-language compilation occurs

using said compilation time interface implemented in said first compiler component, and said compilation time interface implemented in said second compiler component.

5. (Original) The method of claim 4, where said compilation time interface comprises a normalized referencing scheme describing how code in other languages references code compiled by a compiler implementing said normalized referencing scheme.

6. (Original) The method of claim 4, where said compilation time interface comprises data regarding how dependency trees are to be built.

7. (Original) The method of claim 4, where said compilation time interface comprises data regarding the order of compilation of different languages in which code to be compiled has been written.

8. (Original) The method of claim 1, where said prespecified protocol comprises a debugging interface, and where said first framework further provides multi-language debugging according to information from said debugging interface.

9. (Original) The method of claim 1, where said prespecified protocol comprises an error interface, and where said first framework further provides multi-language error handling according to information from said error interface.

10. (Original) The method of claim 1, where said first framework further provides a multi-language design tool for designing code comprising code portions in said first language and code portions in said second language.

11. (Original) The method of claim 10, where said prespecified protocol comprises a design time interface, and where said multi-language design tool uses said design time interface corresponding to a compiler for a given language to control designing of code in said given language.

12. (Previously Presented) The method of claim 10, where said prespecified protocol comprises a background compilation time interface, and where said multi-language design tool uses a background compilation time interface corresponding to a compiler for a given language to control background compilation functionality for code in said given language.

13. (Currently amended) A method for multi-language code editing comprising:

discovering a first design tool that conforms to a set of interfaces included in a prespecified protocol and provides design assistance for code in a first language;

discovering a second design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in a second language, wherein the discovery of the first design tool and the second design tool is effected using an inter-process communication; and

providing a framework which integrates said first design tool and said second design tool to provide multi-language editing, wherein the framework creates a multi-language design tool based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first design tool that conforms to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first design tool and said discovering a second design tool that conforms to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second design tool.

14. (Currently amended) A method for multi-language code editing comprising:

discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said second language, wherein the discovery of the first compiler component, the second compiler component, the first design tool, and the second design tool is effected using an inter-process communication; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality assisting a user in writing code comprising inter-language reference requirements, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first compiler component and a first design tool that conform to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first compiler component and said first design tool and said discovering a second compiler component and a second design tool that conform to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second compiler component and said second design tool.

15. (Currently amended) A method for multi-language code editing comprising:

discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said second language, wherein the discovery of the first compiler component, the second compiler component, the first design tool, and the second design tool is effected using an inter-process communication; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality incorporating background compilation, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first compiler component and a first design tool that conform to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first compiler component and said first design tool and said discovering a second compiler component and a second design tool that conform to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second compiler component and said second design tool.

16. (Currently amended) A method for multi-language compilation comprising:

discovering at least two compiler components that each conform to a set of interfaces included in a prespecified protocol, wherein each of the at least two compiler components provides compilation of code in a corresponding language and the discovery of the at least two compiler components is effected using an inter-process communication; and

providing a framework which integrates each of said at least two compiler components to provide multi-language compilation, wherein the framework creates a multi-language compiler based on the corresponding languages of the at least two compiler components,

where said prespecified protocol comprises a discoverability interface, and where said discovering at least two compiler components that each conform to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said at least two compiler components.

17. (Previously Presented) The method of claim 16, where the inter-process communication is facilitated using a Component Object Model based system.

18. (Previously Presented) The method of claim 16, where said multi-language compilation comprises multi-language compilation to a target platform specified at compilation time.

19. (Currently amended) A computer-readable storage medium for multi-language compilation comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol and provides compilation of code in a second language, wherein the discovery of the first compiler component and the second compiler component is effected using an inter-process communication; and

providing a first framework which uses said first compiler component and said second compiler component to provide multi-language compilation, wherein the framework creates a multi-language compiler based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first compiler component and said discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second compiler component.

20. (Previously Presented) The computer-readable storage medium of claim 19, said instructions further comprising:

discovering a third compiler component which implements said prespecified protocol; and

providing a second framework which integrates said first, second, and third compiler components to provide multi-language compilation.

21. (Canceled)

22. (Previously Presented) The computer-readable storage medium of claim 19, where said prespecified protocol comprises a compilation time interface, and where said provision of multi-language compilation occurs using said compilation time interface implemented in

said first compiler component, and said compilation time interface implemented in said second compiler component.

23. (Previously Presented) The computer-readable storage medium of claim 22, where said compilation time interface comprises a normalized referencing scheme describing how code in other languages references code compiled by a compiler implementing said normalized referencing scheme.

24. (Previously Presented) The computer-readable storage medium of claim 22, where said compilation time interface comprises data regarding how dependency trees are to be built.

25. (Previously Presented) The computer-readable storage medium of claim 22, where said compilation time interface comprises data regarding the order of compilation of different languages in which code to be compiled has been written.

26. (Previously Presented) The computer-readable storage medium of claim 19, where said prespecified protocol comprises a debugging interface, and where said first framework further provides multi-language debugging according to information from said debugging interface.

27. (Previously Presented) The computer-readable storage medium of claim 19, where said prespecified protocol comprises an error interface, and where said first framework further provides multi-language error handling according to information from said error interface.

28. (Previously Presented) The computer-readable storage medium of claim 19, where said first framework further provides a multi-language design tool for designing code comprising code portions in said first language and code portions in said second language.

29. (Previously Presented) The computer-readable storage medium of claim 28, where

said prespecified protocol comprises a design time interface, and where said multi-language design tool uses said design time interface corresponding to a compiler for a given language to control designing of code in said given language.

30. (Previously Presented) The computer-readable storage medium of claim 28, where said prespecified protocol comprises a background compilation time interface, and where said multi-language design tool uses a background compilation time interface corresponding to a compiler for a given language to control background compilation functionality for code in said given language.

31. (Currently amended) A computer-readable storage medium for multi-language code editing comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first design tool that conforms to a set of interfaces included in a prespecified protocol and provides design assistance for code in a first language;

discovering a second design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in a second language, wherein the discovery of the first design tool and the second design tool is effected using an inter-process communication; and

providing a framework which integrates said first design tool and said second design tool to provide multi-language editing, wherein the framework creates a multi-language design tool based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first design tool that conforms to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first design tool and said discovering a second design tool that conforms to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second design tool.

32. (Currently amended) A computer-readable storage medium for multi-language code editing comprising computer executable modules having computer executable instructions,

said instructions comprising instructions for:

discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said second language, wherein the discovery of the first compiler component, the second compiler component, the first design tool, and the second design tool is effected using an inter-process communication; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality assisting a user in writing code comprising inter-language reference requirements, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first compiler component and a first design tool that conform to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first compiler component and said first design tool and said discovering a second compiler component and a second design tool that conform to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second compiler component and said second design tool.

33. (Currently amended) A computer-readable storage medium for multi-language code editing comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering a first compiler component that conforms to a set of interfaces included in a prespecified protocol and provides compilation of code in a first language;

discovering a second compiler component that conforms to the set of interfaces included in said prespecified protocol and provides compilation of code in a second language;

discovering a first design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said first language;

discovering a second design tool that conforms to the set of interfaces included in the prespecified protocol and provides design assistance for code in said second language, wherein the discovery of the first compiler component, the second compiler component, the first design tool, and the second design tool is effected using an inter-process communication; and

providing a framework which integrates said first compiler component, said second compiler component, said first design tool, and said second design tool to provide multi-language design functionality incorporating background compilation, wherein the framework creates a multi-language compiler and a multi-language design tool based on the first language and the second language,

where said prespecified protocol comprises a discoverability interface, and where said discovering a first compiler component and a first design tool that conform to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said first compiler component and said first design tool and said discovering a second compiler component and a second design tool that conform to the set of interfaces included in said prespecified protocol occurs using said discoverability interface implemented in said second compiler component and said second design tool.

34. (Currently amended) A computer-readable storage medium for multi-language code compilation comprising computer executable modules having computer executable instructions, said instructions comprising instructions for:

discovering at least two compiler components that each conform to a set of interfaces included in a prespecified protocol, wherein each of the at least two compiler components provides compilation of code in a corresponding language and the discovery of the at least two compiler components is effected using an inter-process communication; and

providing a framework which integrates each of said at least two compiler components to provide multi-language compilation, wherein the framework creates a multi-language compiler based on the corresponding languages of the at least two compiler components,

where said prespecified protocol comprises a discoverability interface, and where said discovering at least two compiler components that each conform to a set of interfaces included in a prespecified protocol occurs using said discoverability interface implemented in said at least two compiler components.

35. (Previously Presented) The computer-readable storage medium of claim 34, where the inter-process communication is facilitated using a Component Object Model based system.

36. (Previously Presented) The computer-readable storage medium of claim 34, where said multi-language compilation comprises multi-language compilation to a target platform specified at compilation time.

37-45 (Canceled).

REMARKS

Proposed Amendments For Entry by Examiner's Amendment

The examiner contacted Applicant's undersigned representative to indicate that the above-referenced patent application would be allowable if the limitations of claims 3 (and 21) were to be incorporated into each of the independent claims. Applicant has accordingly amended independent claims 1, 13, 14, 15, 16, 19, 31, 32, 33, and 34. Claims 3 and 21 have been canceled. Upon entry of the above amendments, claims 1-2, 4-20, and 22-36 will remain pending. Entry of the proposed amendments and allowance of all pending claims is solicited.

CONCLUSION

In view of the foregoing, Applicants respectfully submit that the claims are allowable and that the present application is in condition for allowance. Entry of the above amendments, reconsideration of the application and a Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow the present application for any reason, the Examiner is encouraged to contact the undersigned attorney, Michael P. Dunnam, at (215) 564-8962 to discuss the resolution of any remaining issues.

Date: July 14, 2009

/Michael P. Dunnam/
Michael P. Dunnam
Registration No. 32,611

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439